Self-Organizing maps for detecting abnormal thermal behavior in data centers

Ignacio Aransay*, Marina Zapater*† Patricia Arroba* and José M. Moya*
*LSI - Electronic Engineering Dpt. - Center for Computational Simulation
Universidad Politécnica de Madrid, Madrid 28040, Spain, {iaransay, parroba, josem}@die.upm.es
†DACYA, Universidad Complutense de Madrid, Madrid 28040, Spain, marina.zapater@ucm.es

***Abstract*—With the advent of Cloud Computing applications and online services, big-scale data center facilities have become economically unsustainable. Reducing the huge expenses is now a priority. However, these cost cut policies are detrimental to the reliability of data centers, reducing the safety margins and increasing the probability of anomalous events, leading to unplanned downtimes. Anomalies in data centers need to be detected in the shortest possible time to mitigate the damage and the economical impact of these downtimes. Our work tackles this problem by proposing the use of topology preserving Artifitial Neural Networks (ANNs) to detect atypical behavior in data centers. ANNs are commonly used for outlier detection, making them a good candidate for clustering and model the normal performance of data centers.**

***Keywords*—anomaly detection, data centers, self-organizing maps**

I. Introduction

Nowadays, with the boom of technology, IT networks and data centers have experienced a rapidly increase, playing the central role in business opportunities and digital services. Data centers have laid the foundation for the development of cutting-edge technologies such as Smart Cities and Internet of Things; in addition to providing the required infrastructure for streamlining customer services like e-commerce and ... For this reason, companies have advanced in virtualization, high-density and hardware design technologies to increase IT resources to ensure availability and quality of the applications offered. Within this context, data centers represent a critical pillar in businesses for companies in a wide range of industries, raising more and more everyday their economic impact in business operations. Their importance is such that 93% of the companies that suffer from a 10-days-outage go bankrupt in less than a year [1].

With so much to be lost, it seems odd that 95% of companies have experienced an unexpected downtime event within the last two years [2]; where its average cost is quantified as $5,600 per minute [3]. Unfortunately, the increase focus on cost reduction and energy efficiency have increased the risk of failures and downtime; increasing operating temperature to save in cooling needs leads to reliability issues. Nonetheless, human factor represents the second leading cause of outages, with a 24% of the total. Human errors have been estimated to cost around $300,000 per incident to the companies surveyed [3]. To mitigate harmufl effects, a full Data Center Infrastructure Manager (DCIM) must be implemented along with regularly tests and defined action protocols. Besides developing policies aiming to reduce anomaly detection time and problem areas for a faster recovery.

In this paper, we propose a solution based on Artificial Neural Networks (ANNs), specifically Self Organizing Maps (SOM), to detect abnormal behavior in the data center moving away from a previously trained reliable performance. The main objective is to provide the data center with clustering techniques to fastly detect the symptoms of an abnormal behavior, which can lead to downtimes. Besides reducing the inactivity period thanks to an early detection.

The key contributions of this paper are the following:

- We present a real time reported clustering technique based on SOMs to detect imminent thermal anomalies in data centers. SOMs make use of temporal and spatial models to widen the scope of the anomalies detected.
- We evaluate the efectiveness of our methodology using traces of anomalies extracted from a real data center. The scenario allows us continuos monitoring of parameters in the facility and controllable generation of anomalies.

The remainder of this paper is organized as follows: previous work on anomaly detection is described in Section II. Section III details the proposed solutions based on clustering techniques. In Section V, the testing scenario is detailed along with the experimental results. Finally, the conclusions and future work are drawn in Section VI.

II. Related Work

Data center anomaly detection researchs have experienced a severe increase with the advent of next-generation business models; pushing data centers to their performance limits.

The techniques used for reducing energy consumption typically clash with the reliability of the infrastructure. Increasing the operating temperature to save in cooling needs leads to a reduction of the Mean Time to Failure (MTTF) of IT equipment [4] and UPS battery life. The Uptime Institute shows in [5] that equipment failure rate doubles for every increase of 10°C above 21°C while reference [6] states that the life of UPS batteries is halved for every 10°C above 25°C. Contrary to this, recent studies [5] states that this effect is smaller than what had been assumed.

The scope and origin of anomalies is extremely wide, yet the state-of-the-art researches have mostly focused on network

security and thermal deviations. Most of the research works for security in networks make use of clustering techniques to detect outliers values. References [7] and [8] examine network traces to detect DoS attacks and worms overflowing by using SOM. Li et al. [9], however, propose detecting anomalies calculating the significance of changes based on the centroid of the IPs within the network. Baldoni et al. [10] developed ANNs to statistically correlate network traffic and power comsumption, to recognize and predict component failures in data centers. These attacks, though common, do not represent the root causes of data center downtimes.

Much research has been done in the area of thermal anomaly detection in data centers. Some approaches opted to use Linear Regression (LR) models not only to detect thermal anomalies [11], but also to rank and prioritize them [12]. Some others, have enhanced thermal efficiency with Computational Fluid Dynamics (CFD) in order to rearrange racks and improve the cooling airflow [13]. Reference [14] implements thermal cameras and correlation models to detect temperature deviations. Besides, they propose a novel thermal-anomaly aware allocation policy to reallocate incoming workload.

Further research works, more similar to ours, make use of Principal Component Analysis (PCA) to detect thermal anomalies in data centers [15]. The use of ANNs is not new either, Yuan et al. [16] propose the implementing hierarchichal ANNs to detect a wide range of temperature anomalies. However, they need to evaluate several metrics such as CPU usage, temperatures of inlet and CPU and fan speed, increasing its complexity.

III. Proposed solution

This section is focused on enumerating the requirements that an anomaly detector focused on data centers must have. Besides detailing our proposed solution to develop a simple but effective anomaly detector to identifiy data center anomalies affecting to the thermal behavior, no matter the causes, with false positive and negative rates.

In the field of data mining, several algorithms have been developed to automatize data processing, something that has been done manually for centuries. Among all the solutions available (support vector machine, density-based techniques, neural networks...) we have considered this last one for detecting anomalous behavior for their flexibility, robustness and self-organization, being ideal for real time applications.

Due to the wide range of space and time temperature anomalies occuring in a data center, it is not feasible to label and classify abnormal data as such. Therefore, the use of supervised learning is not convenient, since we do not have an *a priori* knowledge of the anomaly. On the contrary, unsupervised learning techniques allow us to model and cluster normal performance by finding hidden estimated density patterns based on statistics of the observation; assuming abnormal behavior as a deviation from nomal behavior, no matter the origin and causes.

Furthemore, temperature data need to be quantified to check how new particular samples assemble into the global structure. Therefore, the need of topology based on clustering, to decide whether a particular sample is consistent or not with the previous values within a context.

In this work, we rely on SOM's ability for detecting outliers. Self Organizing Maps provide us the previous mentioned framework. SOMs are clustering algorithms based on ANNs trained using unsupervised learning. Besides, they preserve the topological properties of the input space through the use of neighborhood functions. Among other similar solutions, SOMs provide simple and efficient ways to classify data sets because of its high speed and fast conversion to process real-time data. This makes it a perfect candidate to handle big amounts of data. Besides, its fixed number of nodes is appopriate, since temperature values are within an expected range. This lets us control the quantifiying process and refine the limit between anomaly or not.

Like most of the ANNs, SOMs operate in two phases: training, and mapping. The former consists of the extraction of feature vectors from the data center in order to build the maps. These maps characterize the normal performance of the data center. Once it is done, a mapping phase classifies new feature vectors, comparing their affinity with the clustered data in the maps and labeling them as *normal* or *abnormal*.

These two processes need to be performed periodically to assure the inclusion of new feature vectors in the maps.

A. Training

Our idea is to find temporal and/or spatial inconsistencies in sensed data in order to detect anomalies compromising servers' safety. To this end, we need to carry out a training phase. The aim of this phase is to provide the ANNs with enough information to build an accurate characterization of the realiable performance of servers within the data center, supposing that normal instances are far more frequent than anomalies during the training.

During the training phase, temperature data are extracted and arranged in vectors. The vector size has been set at 3 since it establishes a reasonable commitment between false positive and detection rate; higher n-gram sizes add more sensibility and precision, however, it may be detrimental as it increases the false positive rate.

This process lasts for one day, and it is daily repeated, merging the new data set with the ones we had previously, as explained in III-C. This allows us characterize the average thermal evolution of the facility throughout the day. Vectors are divided in time frames according to different times of the day, characterizing the performance for different time periods, i.e. different demand at different times of the day. Each time frame is represented by a map and it is trained with the vectors corresponding to such time frame.

Maps are formed by a fixed number of nodes spread around the euclidean space (vector size of 3). During the training, vectors are classified according to similarities, by moving the nodes towards their position. In the end, nodes are associated with groups of the input vectors. In other words, input vectors are attached to the associated nodes in the network.

For the training phase of each server, we use two different data sets in order to extend the range of anomalies that can be detected. These sets consist on: i) a temporal set, representing the temporal variation of the internal temperature of a server, and ii) a spatial, representing the external temperature variation

of a server and its neighbors. All servers are trained and tested individually.

1) Implemented Algorithm: Based on the mathematical background of SOM, and considering its algorithm [17], our implementation consists on the following steps:

1) The size of the grid is established. For our puposes, the number of nodes has been set at 8, looking as well for a compromise between detection and false positive rates. Node weights are initialized equidistantly spreading them around the space.
2) A random instance from the training data is extracted.
3) The closest node to the elected instance is established and name Best Matching Unit (BMU).
4) The BMU influence radius is calculated according to Equation 1

$$\sigma(t) = \sigma_0 exp(-\frac{t}{\lambda}), t = 1, 2, 3... \quad (1)$$

where σ_0 is the initial influence radius, λ is a time constant and t the current interation. Influence radius is initially high, decreasing during time according to the time constant.
5) Every node that resides in the σ-neighborhood of the BMU has its weight adjusted according to the following equation 2.

$$W(t+1) = W(t) + L(t) \cdot \Theta(t)(V(t) - W(t)) \quad (2)$$
$$L(t) = L_0 exp(-\frac{t}{\lambda}), t = 1, 2, 3...$$
$$\Theta(t) = exp(-\frac{dist^2}{2\sigma^2(t)}), t = 1, 2, 3...$$

where W(t) and W(t+1) are the weights of the node for time t and t+1 respectively. V(t) is the weight of the extracted instance, and L(t) and Θ(t) are the learning rate and neighborhood function respectively, also decreassing with time. Finally, *dist* stands for the distance between the node being updated and the BMU.
6) Steps 2-5 are repeated for every instance of the training data set.
7) Steps 2-6 are repeated N times until convergence is found.

Temporal and Spatial sets properties are described below.

2) Temporal Maps: Temporal maps let us characterize the performance of servers according to the workload they are executing. For this reason, we have considered as the best parameter the CPU temperature. CPU temperature has a strong relation with the utilization, making it ideal to detect workload execution deviations which may be related to issues in the data center. To reduce the amount of different n-grams, values are quantified in steps of 5°C, without losing relevant information.

Each temporal map is built by extracting feature temporal vectors from each server. These vectors arrange the CPU temperature of servers in n-grams of a time frame window sized 3, corresponding to the (T_{CPU}) of a server for 3 consecutive instants. Equation 3 details the composition of one of these temporal vectors.

$$\text{Temporal vector} \equiv (T_{CPU_t}, T_{CPU_{t-1}}, T_{CPU_{t-2}}) \quad (3)$$

Temporal maps allow us detect which servers are performing workload in a way not previously seen. This can mean host based attacks like resource hogs, or Trojans, idle servers that need to be rebooted or the execution of illegitimate workload.

3) Spatial Maps: Spatial maps are designed to characterize the cooling parameters of the servers. For this reason, we have considered the use of inlet temperatures. In all cases, anomalies affecting refrigeration system have a direct and big impact on the (T_{INLET}) of the servers. Contrary to the previous case, temperature values are not quantified in order to be more precise during the detection.

The map is built by arranging the inlet temperature in n-grams of a space frame windows sized 3, corresponding to (T_{INLET}) of 3 different servers, fitted one above the other within a rack. By adding the height variable, we can detect refrigeration problems that may occur before to the neighbor servers within the rack. These problems are likely to affect, sooner or later, the performance of the whole rack.

$$\text{Spatial vector} \equiv (T_{INLET_{x+1}}, T_{INLET_x}, T_{INLET_{x-1}}) \quad (4)$$

Spatial maps let us control the inlet temperature limits of servers within the data center. A significant increase of this inlet temperature is detected as anomaly in every servers, indicating problems with the refrigeration system, such as CRAC failures.

B. Mapping

The mapping phase consists of detecting unknown behaviors that have not been seen during the training phase, temporal and spatial. We consider anomaly any outlying data that differ a certain threshold from the trained map. When a new feature vector is extracted, it is then assigned to its nearest node (BMU) of the map corresponding to the current time frame. After this, we calculate the euclidean distance from the vector to its corresponding BMU.

During the test, feature vectors not seen in the training appear when temperature sensors start providing data significantly different than before. When this happens, the distance between the feature vector and its corresponding nearest cluster increases, showing evidence of abnormal behavior. Then, any distance greater than a certain threshold distance is declared as an anomaly. To label an feature vector as anomalous, it must be declared as anomaly in the temporal model *OR* the spatial model.

However, this threshold must not be static, since it depends on the variations of the training data. Training data with many variations are likely to be more difficult to cluster, being more spread out over the space. The principal consequence of this is the increase of the error made during the clustering. Therefore, a fixed threshold value may consider normal data as an anomaly, increasing the false positive rate. For this reason, we propose an adaptative threshold, detailed below.

1) Adaptative Threshold: In order to calculate the threshold, we follow this line of thought. Each node has associated a list of vectors, meaning those that have the mentioned node as the Best Matching Unit (BMU). The distance from a vector to the center of the BMU is considered as *clustering error*, since it represents the error made by assuming the center of the BMU as the representative of the vector. Map nodes have circular shape. The maximum error defines the maximum radius that a feature vector seen during the learning process has to its BMU. Distances lesser to this maximum value are considered *normal*, since they are within the radius fixed by the maximum value. Contrary to it, distances greater than the maximum *clustering error* are considered *abnormal* data, as they represent previously unseen values. Therefore, we can establish an adaptative threshold value for each node depending on the maximum distance between a node an its furthest associated vector.

Equation 5 represents this threshold, where the standard deviation is added to decrease the number of false positives, as proposed in [16].

$$Threshold_{NODE} = max(V) + std(V) \tag{5}$$

where V represents the set of feature vectors of the corresponding node.

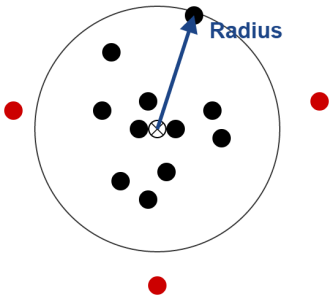

Fig. 1. Adaptative threshold for each node

Figure 1 shows the diagram of the adaptative threshold, applied to every node from the cluster.

C. Retraining phase

Generally, data center workload is not static and it is subject to change. For this reason, the training phase must be repeated after the end of the day, once captured new values which may have appeared, and defininig those appearing the most. The aim of this process is to model a balanced average performance of each server after a long period of time. To do this, the previous training data set of size N and the list of new retrieved values of size N are merged together in a new training data of size N. This new training data is the one used now in the clustering process to rebuild the maps. The process of merging is detailed as follows.

After it, n-grams with the same value are grouped together, recording their number of ocurrences (absolute frequency). To create the new training data, both data sets are merged and averaged by calculating the semisum of the absolute frequencies of n-grams of the same value, after making this assumption: a n-gram that does not exist in one of the sets while being on the other, actually exists but with a frequency of 0. Figure 2 represents on a diagram this merging algorithm.

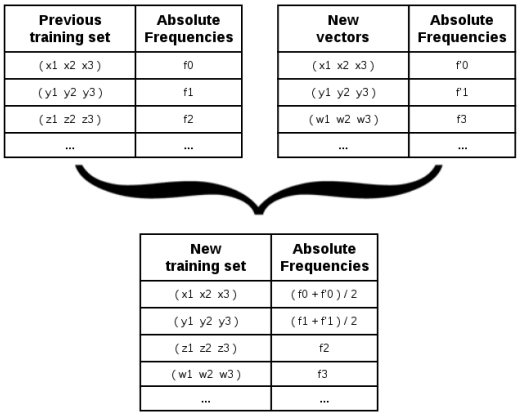

Fig. 2. Merging algorithm for the retraining phase

IV. Experimental setup

In this section, we detail the main aspects of the experiment and the methodology used for detecting anomalies using the proposed solution. Our objective is to ensure that clustering maps are convenient to detect abnormal situations inside a data center, besides offering low false positives rate. For this purpose, we have developed our experiment on a real data room belonging to the research group. Data traces are collected in situ from the data center.

A. Servers

For the sake of clarity, the experiment has been restricted to one rack and three servers, and may be extrapolated to the rest of the data center following the same methodology.

The rack contains 3 servers of two different types of architechture: i) 1 server Dual-Core AMD Opteron SunFire V20z with 4GB of RAM and ii) 2 servers Quad-Core Intel Xeon Fujitsu RX300-S6 with 16GB of RAM.

B. Workload

Servers execute workload emulating real incoming workload with a non-homogeneous Poisson statistical distribution. Workload simulates the demand of a real data center throughout a day, with a time-varying arrival rate. The workload is daily generated by using different tasks of the SPEC CPU 2006 benchmark [18] and scheduled through the SLURM resource manager [19] to the best-fit servers. Thus, each server exectutes its own workload profile.

C. Time frame

Workload executed in the servers depends on the time of the day, having low activity at night and reaching peaks of load in the morning and afternoon. In this situation, we have decided to divide the training data of one day in 4 time frames, both spatial and temporal, corresponding to the natural time divisions of the day, i.e. morning, afternoon, evening and night.

A small time frame can produce high false positive rates due to the daily dissimilarity of the workload executed for each server; incoming workload is modeled as an stochastic process and the SLURM manager is not even. On the other hand, one only time frame would aggregate the performance of the server through the whole day. This aggregation makes impossible to detect anomalies, as the training data is very scattered.

D. Data Collection

Data have been collected from the data room through a self-developed DCIM and monitoring tools. External parameters, as the inlet temperature for each server, are collected through a Wireless Sensor Network (WSN), deployed in the room. Internal parameters, as the CPU temperature for each server, are collected through the Intelligent Platform Management Interface (IPMI) tool. The whole infrastructure lets us monitor the overall data center performance and environmental conditions.

V. Results

A. Detection of CRAC failure

During the training period, the inlet temperature of the servers varies between 23°C and 25°C. CRAC anomalies can be generated by turning off a cooling unit for a certain period of time. By doing this, inlet temperature is expected to increase rapidly. This variation is detected by the spatial model, extracting temperature values that are not previously seen, and can form a threat to the reliability of the room.

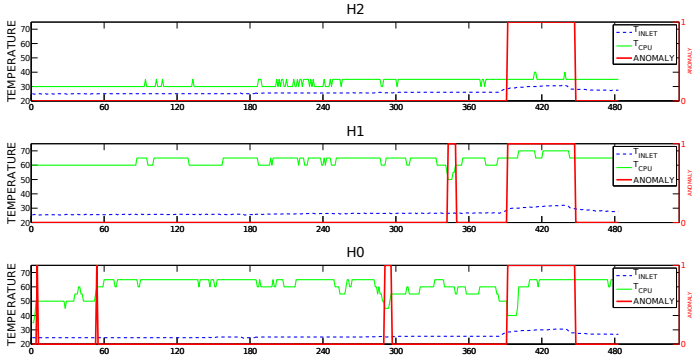

Fig. 3. CRAC anomaly failure affecting the whole rack H0-H1-H2 starting around tick 400

Figure 3 shows the CPU temperature and the inlet temperature for all servers in the rack, coupled with the boolean anomaly expression. Around tick 400, inlet temperature starts increasing, reaching temperature values that clash with the previously seen feature vectors. After 7 ticks, the anomaly is detected by the system, changing the state from low to high in the three servers of the rack at the same time. The spatial model, let us correlate the inlet temperature of servers three by three. When a variation is detected, anomaly is set in all the servers trained with the spatial model, indicating performance problems in a region or zone. The information provided by the inlet temperatures, i.e. the spatial map, is enough to detect CRAC failures with a low detection time of 7 ticks. The CRAC anomaly finishes after 60 ticks, recovering again the previous values and the *normal* state.

Table I. Performance temporal model for each server

Server	Cluster	Time frame	Minimum n-gram	Maximum n-gram
H2	Cluster 0	12:00am - 6:00am	30 30 30	35 35 35
	Cluster 1	6:00am - 12:00pm	30 30 30	35 35 35
	Cluster 2	12:00pm - 6:00pm	30 30 30	35 35 35
	Cluster 3	6:00pm - 0:00am	30 30 30	35 35 35
H1	Cluster 0	12:00am - 6:00am	45 45 45	65 65 65
	Cluster 1	6:00am - 12:00pm	50 50 50	65 65 65
	Cluster 2	12:00pm - 6:00pm	50 50 50	65 65 65
	Cluster 3	6:00pm - 0:00am	45 45 45?	65 65 65?
H0	Cluster 0	12:00am - 6:00am	35 35 35	60 60 60
	Cluster 1	6:00am - 12:00pm	35 35 35	65 65 65
	Cluster 2	12:00pm - 6:00pm	45 45 45	65 65 65
	Cluster 3	6:00pm - 0:00am	35 35 35?	65 65 65?

B. Detection of faulty workload execution

Anomaly in this case does not necessarily mean a problem in the data center, but a rare or unusual activity, which is not expected within such context. This unexpected behavior may occur for several reasons including i) Hardware malfunctions, ii) Resource hogs based attacks, iii) problems based on the resource manager or iv) Computer lock-ups, among others. The main goal of the detector is not to determine the origin of these activity, but to immediately recognize them and reduce the response time, if necessary.

1) Ideal executed workload: First, we need to have an average real behavior model of the performance of each server. For this to be done, each server runs an instance of the anomaly detector during one week. As explained in section IV-B, incoming workload is modeled depending on the time of the day. This incoming workload has a defined profile during the day, however, the host allocation depends on several factors. Therefore, the temporal profile that each server has is not the same every day, but it has a similar tendency. Therefore, after running the algortihm during one week, being retrained every day, it is expected to have an overall model of real performance for each server.

Table I shows the results of the clustering process after one week of execution. In it, it can be seen the minimum and maximum temporal n-gram values reached during each time frame.

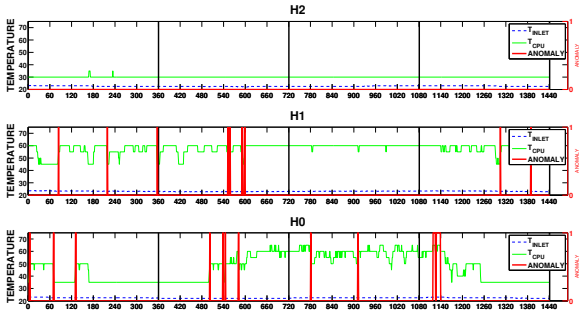

Fig. 4. One day representative workload

Figure 4 shows the temperatures of the three servers during one representative day, where the timeline is divided in four,

corresponding to the four cluster daily divisions. On it, we assume that there are not anomalies, however, some ticks have been classified as such. By counting the number of ocurrences, we can give an approximate representative rate of false positives. In *H0*, there are 34 ticks set as abnormal out of the 1440 total (1 tick per minute during one day). Therefore, the false positive rate for *H0* is 2.36%. Following the same idea, the false positive rate for *H1* is 0.83% and 0% for *H2*.

2) Generation of anomaly: Detecting workload execution anomalies in data centers is not easy, since the workload tends to be highly variable. This variation is detected by the temporal model, individually detecting misbehaving servers.

To generate an anomaly in this section, we force a workload variation by increasing the assigned resources to one particular node, and so its CPU temperature or by causing an idle state which is not supposed to occur during an specific time frame.

Figure 5 shows the results provided by the anomaly detector caused by an abnormal idled performance of H1 during the time frame 2, from 12:00pm to 6:00pm. Around tick 320, H1, that until there was having a normal operation within the margins of the training model, according to Table I, goes into *idle* mode. After 1 tick, the detector extracts new vectors corresponding to CPU temperature of 45C, that had not appear during the training phase. In this moment, the detector immediately changes the state of anomaly from 0 to 1. After tick 620, H1 recovers to its normal performance, changing again the state of the anomaly indicating the proper operation. Besides this, around ticks 90 and 190 in H2 and 65, 80, 440, 510 and 520 in H0 anomalies have also been detected. However, they are sporadic anomalies, typically related to sudden variations in the CPU temperature. Since they represent specific moments when a change of state occurs, they can be defined as false positives.

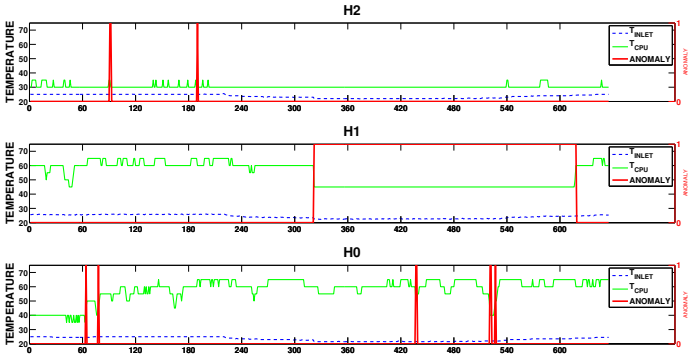

Fig. 5. Contextual anomaly in H1 starting around tick 320

Figure 6 shows the results of the anomaly detector when an illegitimate workload is executed in H0 during a time frame when it should not, i.e. Cluster 0, from 12:00am to 6:00am. Around tick 65, the server H0 starts giving abnormal temperature values of 75C during a long-lasting period of time, indicating the presence of abnormalities. This misbehavior is captured by the anomaly detector, changing the state of anomaly from low to high. Around tick 110, the workload misconfiguration dissapears, and so the anomaly. As in the previous case, false positive rates are present in ticks 400 and 460 for H0 and in ticks 105 and 110 for H1.

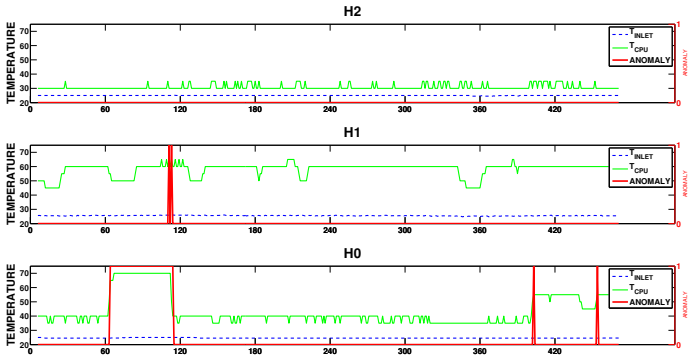

Fig. 6. Contextual anomaly in H0 starting around tick 65

VI. Conclusions and Future Work

Detecting abnormal perfomance in data centers is imperative to allow the rapid actuation upon data center anomalies, since any error decting them can be reponsible of big amount of losses and even bankruptcies. The work presented in this paper makes contributions in the area of anomaly detection with a clustering methodology based on SOM. The use of SOM allows us identify thermal variations in data centers. Since every server is running a temporal and a spatial instance, we are able to differenciate between problems affecting one area or region with the spatial model and workload misconfiguration in specific nodes with the temporal model. Moreover, we have achieved low detection times for both CRAC malfunction and workload misconfiguration, which is essential to minimize the economic effects of a downtime in the facility.

Further research is being done for coupling anomaly detection systems with a Trust and Reputation System, to develop a robust to anomalies allocation policy, depending on an overall grade, called reputation, to maximize reliability by detecting and isolating malfunctioning hosts

Acknowledgment

This project has been partially supported by the Spanish Ministry of Economy and Competitiveness, under contracts TEC201233892, IPT20121041430000 and RTC201427173.

References

[1] *Surviving Downtime in the Datacenter*, 2013.

[2] P. Institute, "Addressing the leading root causes of downtime," Ponemon Institure sponsored by Emerson Network Power, Tech. Rep., 2010.

[3] ——, "Understanding the cost of data center downtime," Ponemon Institure sponsored by Emerson Network Power, Tech. Rep., 2011.

[4] D. Atienza, G. De Micheli, L. Benini, J. L. Ayala, P. G. Del Valle, M. DeBole, and V. Narayanan, "Reliability-aware design for nanometer-scale devices," in *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*. IEEE, 2008, pp. 549–554.

[5] R. Menuet and W. P. Turner, "Continuous cooling is required for continuous availability," 2006.

[6] U. S. plc., "A guide to ensuring your ups batteries don't fail from ups systems," http://www.upssystems.co.uk/knowledge-base/the-it-professionals-guide-to-standby-power/part-8-how-to-ensure-your-batteries-dont-fail/, 2015, accessed: 2015-05-18.

[7] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," in *Recent Advances in Intrusion Detection*. Springer, 2003, pp. 36–54.

[8] K. Labib and R. Vemuri, "Nsom: A real-time network-based intrusion detection system using self-organizing maps," *Networks and Security*, pp. 1–6, 2002.

[9] A. Li, L. Gu, and K. Xu, "Fast anomaly detection for large data centers," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[10] R. Baldoni, A. Cerocchi, C. Ciccotelli, A. Donno, F. Lombardi, and L. Montanari, "Towards a non-intrusive recognition of anomalous system behavior in data centers," in *Computer Safety, Reliability, and Security*. Springer, 2014, pp. 350–359.

[11] B. Haaland, W. Min, P. Z. Qian, and Y. Amemiya, "A statistical approach to thermal management of data centers under steady state and system perturbations," *Journal of the American Statistical Association*, vol. 105, no. 491, pp. 1030–1041, 2010.

[12] K. Viswanathan, L. Choudur, V. Talwar, C. Wang, G. Macdonald, and W. Satterfield, "Ranking anomalies in data centers," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 2012, pp. 79–87.

[13] R. Romadhon, M. Ali, A. M. Mahdzir, and Y. A. Abakr, "Optimization of cooling systems in data centre by computational fluid dynamics model and simulation," in *Innovative Technologies in Intelligent Systems and Industrial Applications, 2009. CITISIA 2009*. IEEE, 2009, pp. 322–327.

[14] E. K. Lee, H. Viswanathan, and D. Pompili, "Model-based thermal anomaly detection in cloud datacenters," in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*. IEEE, 2013, pp. 191–198.

[15] M. Marwah, R. Sharma, W. Lugo, and L. Bautista, "Anomalous thermal behavior detection in data centers using hierarchical pca," *SensorKDD in conjunction with KDD*, 2010.

[16] Y. Yuan, E. K. Lee, D. Pompili, and J. Liao, "Thermal anomaly detection in datacenters," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 226, no. 8, pp. 2104–2117, 2012.

[17] T. Kohonen, "Som toolbox: Intro to som," http://www.cis.hut.fi/projects/somtoolbox/theory/somalgorithm.shtml, 2005, accessed: 2015-06-10.

[18] J. L. Henning, "Spec cpu2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 4, pp. 1–17, 2006.

[19] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing*. Springer, 2003, pp. 44–60.